AMENDMENTS TO THE CLAIMS

In accordance with the PTO's amendment format, a detailed listing of all claims has been provided. A status identifier is provided for each claim in parentheses following each claim number.

In the Claims:

Please amend claims 5, 6, and 21.

Claims 1-5, 11-13, and 15-20 are allowed.

Claim 14 is previously canceled.

No new claims are added.

Claims 1-13, 15-25 are currently pending.

Listing of the Claims

1. (Allowed) A method for debugging a target computer that utilizes virtual memory paging, the method comprising:

transferring physical memory data from the target computer to a host computer; and

replicating virtual memory data from the physical memory data on the host computer, wherein the virtual memory data on the host computer is identical to virtual memory data on the target computer.

2. (Allowed) The method as recited in claim 1, further comprising debugging a fault on the target computer by analyzing replicated data on the host computer.

3. (Allowed) The method as recited in claim 1, further comprising caching the replicated data in memory on the host computer.

4. (Allowed) The method as recited in claim 1, wherein the target computer includes an operating system that uses table-driven paged memory management.

5. (Currently Amended) The method as recited in claim 1, wherein:

the target computer includes a processor that has halted execution; and

the virtual memory data is located in physical memory of the target computer.

6. (Currently Amended) A host computing system, comprising:

a processor;

memory;

means for establishing a connection between the memory and memory of a target computer;

a data retrieval component configured to transfer address data from memory of the target computer to the memory;

an address translation component configured to replicate virtual memory addresses from the address data in the memory[[,]] ; and

wherein the virtual memory ~~addresses~~ data in the host computing system are identical to virtual memory ~~addresses~~ data of the target computer.

7. (Original) The host computing system as recited in claim 6, further comprising cache memory configured to store the replicated virtual memory addresses.

8. (Original) The host computing system as recited in claim 6, wherein the host-side address translation component is further configured to validate the replicated virtual memory addresses.

9. (Original) The host computing system as recited in claim 6, further comprising a memory management verifier that verifies that a processor of the target computing system has memory management enabled.

10. (Original) The host computing system as recited in claim 6, wherein the means for establishing a connection between the memory and memory of a target computer comprises hardware-assisted debug probes.

11. (Allowed) A method, comprising:

accessing address tables from physical memory of a target computer system;

replicating the address tables on a host computing system; and

using data contained in the address tables to derive virtual address data that was used on the target computer system, wherein the virtual address data on the host computer system are identical to virtual address data on the target computer system.

12. (Allowed) The method as recited in claim 11, further comprising storing the address tables in memory on the host computer system.

13. (Allowed) The method as recited in claim 11, further comprising caching the virtual address data on the host computer system.

14. (Canceled)

15. (Allowed) The method as recited in claim 11, further comprising determining if memory management of a target computer system processor is enabled.

16. (Allowed) The method as recited in claim 11, further comprising performing the method only if memory management of a target computer system processor is enabled.

17. (Allowed) The method as recited in claim 11, wherein the accessing further comprises:

locating the address tables in physical memory of the target computer system; and

reading the address tables from the target computer.

18. (Allowed) The method as recited in claim 11, further comprising validating the virtual address data to ensure it is identical to the virtual address data stored on the target computer system.

19. (Allowed) The method as recited in claim 11, further comprising debugging a fault that occurred on the target computer by analyzing the virtual address data on the host computer system.

20. (Allowed) A computer-readable medium containing processor-executable instructions that, when executed on a processor, perform the method of claim 11.

21. (Currently Amended) One or more computer-readable media containing computer-executable instructions that, when executed by a computer, perform the following steps:

transferring physical memory data of a target computer to a host computer;

translating address data contained in the physical memory data to virtual addresses utilized by the target computer[[,]] and

wherein the virtual ~~addresses~~ memory data on the host computer are identical to virtual ~~addresses~~ memory data on the target computer.

22. (Original) The one or more computer-readable media as recited in claim 21, further comprising computer-executable instructions that, when executed by a computer, perform the following steps:

locating address data in the physical memory of the target computer; and

transferring only the address data to the host computer.

23. (Original) The one or more computer-readable media as recited in claim 21, further comprising computer-executable instructions that, when executed by a computer, caches data transferred from the target computer on the host computer.

24. (Original) The one or more computer-readable media as recited in claim 21, further comprising computer-executable instructions that, when executed by a computer, validating the transferred data to determine if the transferred data is identical to the contents of the physical memory.

25. (Original) The one or more computer-readable media as recited in claim 21, further comprising computer-executable instructions that, when executed by a computer, determining if memory management is enabled on a processor in the target computer prior to transferring data.